Power Strip Packing of Malleable Demands in Smart Grid

Mohammad M. Karbasioun*, Gennady Shaikhet†, Evangelos Kranakis‡, Ioannis Lambadaris*
Carleton University, Ottawa, ON, Canada
*Department of Systems and Computer Engineering {mkarbasi, ioannis}@sce.carleton.ca
†School of Mathematics and Statistics {gennady}@math.carleton.ca
‡School of Computer Science {kranakis}@scs.carleton.ca

arXiv:1302.3889v1 [cs.DS] 15 Feb 2013

***Abstract*—We consider a problem of supplying electricity to a set of $\mathcal{N}$ customers in a smart-grid framework. Each customer requires a certain amount of electrical energy which has to be supplied during the time interval $[0, 1]$. We assume that each demand has to be supplied without interruption, with possible duration between ℓ and r, which are given system parameters ($\ell \leq r$). At each moment of time, the power of the grid is the sum of all the consumption rates for the demands being supplied at that moment. Our goal is to find an assignment that minimizes the $power\ peak$ - maximal power over $[0, 1]$ - while satisfying all the demands. To do this first we find the lower bound of optimal power peak. We show that the problem depends on whether or not the pair ℓ, r belongs to a "good" region $\mathcal{G}$. If it does - then an optimal assignment almost perfectly "fills" the rectangle $time \times power = [0, 1] \times [0, A]$ with A being the sum of all the energy demands - thus achieving an optimal power peak A. Conversely, if ℓ, r do not belong to $\mathcal{G}$, we identify the lower bound $\overline{A} > A$ on the optimal value of power peak and introduce a simple linear time algorithm that almost perfectly arranges all the demands in a rectangle $[0, A/\overline{A}] \times [0, \overline{A}]$ and show that it is asymptotically optimal.**

I. Introduction

A. Motivations

One of the main goals of Demand Side Management (DSM) in smart grid is to reduce the peak to average ratio (PAR) of the total consumed power of the network [1]. It has several benefits for the grid such as reducing the amount of additional supplementary power needed to satisfy the demand during peak hours which itself results in decreasing the CO_2 emissions of power plants. Also reducing the PAR decreases the possibility of power outage due to sudden increase of demands. Furthermore with the expected presence of plug-in hybrid electric vehicles (PHEVs) to the market, finding a proper scheduling of the electric demands becomes even more crucial, since during charging hours of PHEVs the average household load is expected to be doubled [2]. Also by increasing the number of PHEVs, it is vital to design and deploy suitable charging stations with proper scheduling scheme for serving the customers [3]. Therefore finding a proper scheduling algorithm resulting in a suitable PAR is very important in the smart grid. Different approaches and schemes are defined in this field, such as direct load control (DLC) [4] (in which the utility can directly control the energy consumption of some loads in customer side) and real time pricing (RTP) [5] (which propose different pricing schemes during a day). In addition in [6] authors propose a distributed algorithm using game theory. In their model the power network consists of demands each with its own energy demand and also with its own minimum and maximum acceptable power level and should be scheduled in their own requested time intervals. Even though the goal of their algorithm is reducing the total convex cost of the network, they still show that their algorithm can be used to reduce the PAR in the network. The problem in [6] is *off-line* Scheduling of demands, since complete knowledge of the demands, such as number of demands and the amount of energy needed by each of them are known in advance. Other scenarios can happen when we don't have complete knowledge of the demands and hence we need to perform *on-line* scheduling. For example in [7] authors try to find some algorithms for different levels of knowledge about the demands such as arrival times, durations and power intensities. Also in [8] and [9] it is attempted to devise on-line scheduling for the random arrival of demands. In this paper we focus on off-line scheduling with a complete knowledge of the demands prior to performing scheduling.

B. Model and Related Works

Consider a set $\mathcal{N} = \{1, 2, ..., n\}$ of energy demands $\{A_i, i \in \mathcal{N}\}$, needed to be scheduled in a finite time interval $[0, 1]$. We assume that only "rectangular" shape of scheduling is permitted, meaning that each demand $i \in \mathcal{N}$ has to be supplied without interruption in some interval $[\tau_i, \tau_i + s_i]$ with a constant power intensity $d_i = \frac{A_i}{s_i}$. Obviously,

$$0 \leq \tau_i \leq \tau_i + s_i \leq 1, \qquad i \in \mathcal{N}. \tag{1}$$

In addition to (1), we impose a *demand malleability* constraint. That is, we assume the power system has parameters ℓ and r with $0 \leq \ell \leq r \leq 1$ so that

$$\ell \leq s_i \leq r, \qquad i \in \mathcal{N}. \tag{2}$$

This is motivated by existence electric appliances with flexibility on the charging rate. As a special application we may think of a PHEV parking lot with charging facilities where all customers either are known a priori or should register before parking their vehicles.

A set of pairs $\pi = \{(\tau_i, s_i), i \in \mathcal{N}\}$, satisfying (1)–(2) for given parameters (ℓ, r), will be called a *scheduling policy*. Let $\Pi = \Pi^{(\ell,r)}$ be a set of all policies.

For a policy $\pi \in \Pi$, we define its maximal power as

$$P^{\pi}_{max} = \max_{t\in[0,1]}\{P^{\pi}(t)\}, \tag{3}$$

where, for $(\tau_i, s_i) \in \pi$,

$$P^{\pi}(t) = \sum_{i=1}^{n}\left(\frac{A_i}{s_i}\cdot 1_{\{\tau_i\le t\le\tau_i+s_i\}}\right),\ \ 0 \le t \le 1.$$

We are interested in finding a scheduling that minimizes the peak to average ratio in the power grid, i.e., in finding

$$P_{opt} = \inf_{\pi\in\Pi} P^{\pi}_{max}. \tag{4}$$

Note, that P_{opt}, although it is not stated explicitly, depends on the parameters ℓ and r.

C. Related literature

Our setting resembles a so-called strip-packing problem [10] and [11]. Indeed by viewing the demands as rectangles, we want to pack them with their side s_i parallel to horizontal axis in a rectangular bin of width $[0, 1] \times P_{opt}$ where an optimal height P_{opt} is unknown. The problem is known to be NP complete (see [12]) and therefore an optimal height cannot always be computed in polynomial time.

In the traditional strip packing (TSP) problem, however, the height at any time t ($H^{\pi}(t)$) for a scheduling policy π is defined as the *uppermost boundary* of scheduled rectangles at time t, while in our model (power strip packing - PSP) the height of the strip packing is obtained from Equation (3). This difference arises from the nature of the electric power, in which the overall height (i.e. power) at any given time is the sum of the scheduled (i.e. active) demands (See Figure 1). Naturally, for any policy π, $P^{\pi}(t) \le H^{\pi}(t)$ at any time t and hence $P^{\pi}_{max} \le H^{\pi}_{max}$.

For traditional strip packing problem, most commonly referred work is [11], which introduces an algorithm AFPTAS, whose performance $AFPTAS(\mathcal{N})$ satisfies $AFPTAS(\mathcal{N}) \le (1+\epsilon)H_{opt} + O(1/\epsilon^2)$. The algorithm is polynomial in both n (the number of demands) and $(1/\epsilon)$ where the running time is $O(n\log n + \log^3 n\epsilon^{-6}\ \log^3\epsilon^{-1})$. Using AFPTAS, Jansen [13] generalized the setting of [11] to *"Malleable Tasks"* where each task could use different number of resources (e.g. processor, memories,...) which can also alternate their service times. Both algorithms, especially the one in [13], are complex. In our case, however, a much simpler, linear-time algorithm is used to obtain a scheduling policy π with a performance P^{π}_{max} satisfying

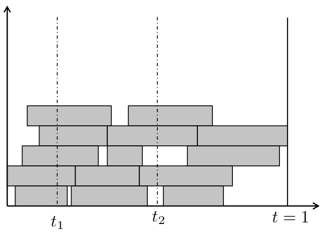

Fig. 1: Different interpretations of power and height in PSP and TSP. For simplicity the height of each rectangle is assumed to be 1. In PSP: $P(t_1) = 5$ and $P(t_2) = 3$ where in TSP $H(t_1) = H(t_2) = 5$.

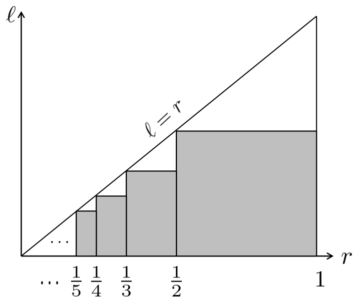

Fig. 2: Good region vs. Bad region. The Good region is shown with shadow.

$$P^{\pi}_{max} \le P_{opt} + \frac{A_{max}}{\ell}, \quad A_{max} = \max_{i\in\mathcal{N}} A_i; \tag{5}$$

which, using the language of [11] and [13], corresponds to performance ratio being exactly 1 (as an opposite to $1+\epsilon$).

In next section we will present Theorem 1 which is the main result of the paper. Then in Sections III and IV we analyse the problem to find proper lower bound and upper bound for P_{opt}. Then based on theses results in Section V we will prove Theorem 1. Finally in Section VI we will present some computational results.

II. Main results

We start with some preparatory work. Let ℓ and r be fixed. A real number $w > 0$ is called **achievable**, if it can be represented as an integer combination of numbers from the interval $[\ell, r]$. That is, if there exist an integer number q and a set of positive real values: $s_1, ..., s_q \in [\ell, r]$ such that $\sum_{i=1}^{q} s_i = w$.

Lemma 1: A value w is achievable if and only if $\lceil \frac{w}{r} \rceil \le \frac{w}{\ell}$. The proof is given in Appendix A.

Lemma 1 implies that the point $w = 1$ is achievable if $\lceil \frac{1}{r} \rceil \le \frac{1}{\ell}$. Define a **good region** to be the set of all pairs (ℓ, r) that makes $w = 1$ an achievable point (see Fig. 2):

$$\mathcal{G} = \{(\ell, r),\ \ell \le r \text{ and } 1 \text{ is achievable}\}.$$

If w is not achievable, we define *its largest achievable point* $w^* = w^*(w)$ as:

$$w^* = \sup\{v\ : v < w, \text{ such that } v \text{ is achievable}\}$$

Lemma 2: If w is not achievable, then $w^*(w) = r \cdot \lfloor \frac{w}{r} \rfloor$. The proof is given in Appendix B.

Using Lemma 2, if $(\ell, r) \notin \mathcal{G}$, then $Z^* = Z^*(1) = r \cdot \lfloor \frac{1}{r} \rfloor$. We are ready to state the main result of the paper.

Theorem 1: Let $A = \sum_{i=1}^{n} A_i$. Then $P_{opt} \in [\overline{A},\ \overline{A} + \frac{A_{max}}{\ell}]$, where

$$\overline{A} = \begin{cases} A & ,\ \text{if } (\ell, r) \in \mathcal{G}, \\ \dfrac{A}{Z^*} & ,\ \text{if } (\ell, r) \notin \mathcal{G}. \end{cases} \tag{6}$$

As the number of demands n grows, the value of $\overline{A}$ grows as well, thus making the interval $\left[\overline{A},\ \overline{A} + \frac{A_{max}}{\ell}\right]$ tight. Therefore, every scheduling policy π, whose corresponding value $P^{\pi}_{\max}$ belongs to that interval, will be considered asymptotically optimal. In the last section two of such policies will be introduced.

III. Lower bound of P_{opt}

Theorem 1 will be proved in two main steps, each dealing with lower and upper bounds respectively. Here we discuss about lower bound.

Lemma 3:

$$P_{opt} \geq \overline{A}. \tag{7}$$

Proof: The Inequality (7) is obvious for $(\ell, r) \in \mathcal{G}$ (making $Z^* = 1$ and $\overline{A} = A$). Therefore, in what follows we assume $(\ell, r) \notin \mathcal{G}$ and hence $Z^* < 1$. To prove Lemma 3 we use the concept of the *Fractional Strip Packing (FSP)* [14], according to which it is allowed to perform horizontal cutting to each rectangle (s_i, d_i) in order to get a set of equal width rectangles $\{(s_i, d_{i1}), (s_i, d_{i2}), \ldots, (s_i, d_{iQ_i})\}$ such that $\sum_{j=1}^{Q_i} d_{ij} = d_i$ for all $i = 1, \ldots, n$. Then, instead of packing the original set $\{(s_i, d_i) \quad i = 1, \ldots, n\}$ of rectangles, a packing of a newly obtained set is performed.

Assume a certain set $\mathcal{N}$ of demands is given. For any FSP policy θ, (that includes both assigning (s_i, d_i), $i \in \mathcal{N}$, as well as cutting), let P^{θ}_{max} and H^{θ}_{max} denote the maximal achieved heights, calculated according to traditional and power strip packing respectively. And let also P^{F}_{opt} and H^{F}_{opt} denote the FSP optimal values for traditional and power strip packing. Clearly, $A \leq P^{F}_{opt} \leq H^{F}_{opt}$ and also

$$A \leq P^{F}_{opt} \leq P_{opt} \ , \tag{8}$$

$$A \leq H^{F}_{opt} \leq H_{opt} \ . \tag{9}$$

Next we will show that $\frac{A}{Z^*} \leq P^{F}_{opt}$, which, combined with (8), proves Lemma 3.

Assume an arbitrary scheduling policy $\pi \in \Pi^{(\ell,r)}$ is given. Cut each demand (s_i, d_i) horizontally to get the new set of very short height rectangles $\mathcal{N}' = \{(s_i, d_{i1}), (s_i, d_{i2}), \ldots, (s_i, d_{iQ_i})\}$ such that $d_{ij} = \delta \ \ \forall i, j$ where δ is an infinitesimal positive real value. Now we try to pack these narrow rectangles in a strip of width 1 as follows: Let call an FSP policy θ^* as *filling* when there will not remain space to put any new rectangle with any arbitrary width s_i ($\ell \leq s_i \leq r$) in each of its rows, i.e. the sum of all gaps in each row is always less than ℓ. (maybe except for the last row). (Θ^* is the set of all possible fillings θ^*). Now to pack all the narrow rectangle with a *filling*, we start with the first narrow rectangle and put it in the first row and then try to *fill* this row with subsequent narrow rectangles. After filling a row we continue with filling subsequent rows until all the narrow rectangles will be scheduled.

Lemma 4:

$$P^{F}_{opt} = \min_{\theta^* \in \Theta^*} P^{\theta^*}_{max} \tag{10}$$

The proof is given in Appendix C. So with respect to Equation (10), in the sequel to show that $\frac{A}{Z*} \leq P^{F}_{opt}$ we just consider fillings $\theta^* \in \Theta^*$.

From Lemma 2 we know that in each row we cannot cover the time axis more than Z^*, then the maximum height (i.e. the upper boundary of the top row) cannot be less than $\frac{A}{Z^*}$, otherwise the total area will be less than A. Combining this fact with Inequality (9) results in:

$$\frac{A}{Z^*} \leq H^{F}_{opt} \leq H_{opt} \tag{11}$$

It proves Lemma 3 for TSP, but to prove this Lemma for PSP we should take into account all possible *gaps* in each row. Because, as mentioned before, the way of computing the maximum height in PSP is different from TSP. Indeed in PSP in calculating the height or more precisely the total power at each time, only the active demands at that time will be considered and *gaps* are not taken into account (Equation (3) and Figure 1). Also note that by *filling* each row arbitrarily, these gaps can split in more than just one part in a row and may happen in different places of the time axis for different rows. Lemma 5 tells that in any arbitrary *filling* there will be exactly $K_0 = \lfloor \frac{1}{r} \rfloor$ rectangles in each row and then in Lemma 6 it will be shown that for any arbitrary *filling*, there will be exactly K_0 equal length and identically placed *"gap-free"* interval in each row. Therefore for each filling θ^*, the resulted P_{max} here is indeed equal to H_{max} and consequently $P^{F}_{opt} = H^{F}_{opt}$. Eventually using Lemma 4 together with (11) proves Lemma 3 for PSP.

Lemma 5: In any filling θ^* for a set of narrow rectangles of arbitrary widths s_i (s.t. $\ell \leq s_i \leq r$), exactly $K_0 = \lfloor \frac{1}{r} \rfloor = \lfloor \frac{1}{\ell} \rfloor$ narrow rectangles are needed (This may not apply for the last row which may contain less rectangles).

The proof is given in Appendix D.

Lemma 6: Under the same conditions as in Lemma 5, each row will contain K_0 identical (i.e. equal length and identically placed) intervals, $(1 - (K_0 - i + 1) \cdot \ell \ , \ i \cdot \ell)$ for all $i = 1, \ldots, K_0$, which cannot contain any gaps (maybe except for the last row).

Proof: Assume that we are trying to put K_0 arbitrary width narrow rectangles in a row. Now assume that we want to put the i^{th} rectangle in this row. The largest value for the starting time of i^{th} narrow rectangle, i.e. τ_i, occurs when all the narrow rectangles starting after τ_i, including i^{th} rectangle itself, have the same width ℓ. Also all possible gaps happen before τ_i. Then the length of the remaining time after τ_i is : $(K_0 - (i - 1)) \cdot \ell$ and hence the largest value for τ_i is as follows:

$$\tau_{i,largest} = 1 - (K_0 - (i - 1)) \cdot \ell$$

On the other hand the smallest value for the finishing time of this rectangle, i.e. $f_i = \tau_i + s_i$, happens only when this rectangle and all of the previous $(i - 1)$ narrow rectangles have the same width ℓ and no gap placed in the time axis until after the f_i. Then the smallest value for f_i is:

$$f_{i,smallest} = i \cdot \ell$$

Therefore i^{th} narrow rectangle starts at most at time $\tau_{i,largest}$ and remains active at least until $f_{i,smallest}$. Then we conclude that the i^{th} narrow rectangle (and **only** this one) is definitely active during the following time interval $(1-(K_0-i+1)\cdot\ell, i\cdot\ell)$ with the length:

$$\begin{aligned} &i\cdot\ell-(1-(K_0-i+1)\cdot\ell) \\ =&(K_0+1)\cdot\ell-1=\left\lceil\frac{1}{\ell}\right\rceil\cdot\ell-1>0 \end{aligned}$$

Therefore we have $K_0=\lfloor\frac{1}{r}\rfloor=\lfloor\frac{1}{\ell}\rfloor$ active intervals with equal non-zero length $(K_0+1)\cdot\ell-1$. ■

Note that it is impossible for a demand to be completely placed in the interval between two consecutive active intervals, because the length of such an interval is $1-K_0\cdot\ell$ which is less than ℓ (because $1<\lceil\frac{1}{\ell}\rceil=(K_0+1)\cdot\ell$).

Corollary 1: In i^{th} active interval of any filling θ^* (Lemma 6), when filling each row with narrow rectangles, definitely the i^{th} narrow rectangle in that row is scheduled and the i^{th} narrow rectangle is the only one which can be scheduled in this interval (maybe except for the last row).

Using Lemma 6, we conclude that in FSP when *filling* the rows with any arbitrary placement and with any arbitrary width (s_i) for narrow rectangles, there will be exactly K_0 gap-free *identical* intervals in every row. Therefore the $P^{\theta^*}_{max}$ is exactly the same as $H^{\theta^*}_{max}$ which is the sum of the heights of all the rows.

Note that no one can change FSP above and get the lower $P^{\theta^*}_{max}$. For example if one tries to decrease the $P^{\theta^*}_{max}$ by just one level, it needs to remove at least K_0 narrow rectangles from different gap-free intervals, either from one row or K_0 different rows (corollary 1). Then these narrow rectangles will form at least one new row which then only can keep the total height the same as before and cannot decrease the $P^{\theta^*}_{max}$. So $P^{\theta^*}_{max}$ above is in fact minimum among all possible *fillings* and hence using (10) in Lemma 4 it is equal to P^F_{opt}.

From Lemma 2 we know that in a filling of narrow rectangles, in each row we cannot cover the time axis more than $Z^*=r\cdot\lfloor\frac{1}{r}\rfloor$, then the maximum height (i.e. the upper boundary of the top row) cannot be less than $\frac{A}{Z^*}$, otherwise the total area will be less than A. Therefore we have: $\frac{A}{Z^*}\le P^F_{opt}\le P_{opt}$ which proves Lemma 3. ■

In the next section we will use a simple linear time algorithm, in which we try to keep the $P(t)$ (and hence P_{max}) around $\frac{A}{Z^*}$. Using this algorithm we will show that the $P_{opt}\le\overline{A}+\frac{A_{max}}{\ell}$ which proves the upper bound of Theorem 1 and hence it also proves that this algorithm is asymptotically optimum.

IV. Algorithm

From Lemma 1, we know that $t=1$ is achievable if and only if $\lceil\frac{1}{r}\rceil\le\frac{1}{\ell}$. Based on this fact, in the sequel, we will introduce different cases and in each case we will find a proper policy which results in acceptable P_{max}.

A. Ideal Cases

First assume that $\ell\le1$ and $r=1$ (So 1 is achievable). In this case we have an optimal policy π^* as follows: stretch the width of each demand to the whole time interval $[0,1]$, and then piling the demands up on top of each other. Another ideal case is when we have $\ell\le\frac{A_i}{A}\le r$, $i=1,\cdots,n$. In this case in optimal policy π^* we have: $s_i=\frac{A_i}{A}$, therefore since $\sum_{i=1}^n s_i=\sum_{i=1}^n\frac{A_i}{A}=1$, 1 is achievable. So in this case we can simply stretch each intensity to A as $A_i=\frac{A_i}{A}\times A$ and place the demands side by side. In both cases we have $P(t)=A\quad$ for all $t\in[0,1]$ and hence:

$$P^{\pi^*}_{\max}=A\le\overline{A}+\frac{A_{max}}{\ell} \tag{12}$$

So for Ideal cases, we can pack the demands in a strip of height A and and because $A\le P_{opt}\le H_{opt}$, the output in this case is absolutely optimum.

B. Near Ideal Cases

In this case $t=1$ is achievable and hence $Z^*=1$. Then we perform scheduling policy π as follows: Simply divide the time interval $[0,1]$ into $K_0=\lceil\frac{1}{r}\rceil$ non-overlapping time slots with equal length $S_0=\frac{1}{K_0}$ and since in this case we have $\lceil\frac{1}{r}\rceil\le\frac{1}{\ell}$ (Lemma 1), we are sure that $\ell\le S_0\le r$.

Now we can start with the first demand and put it in the first slot and continue this until the height of that slot becomes greater or equal to the threshold value, A, then continue by putting the next demand in the next slot and doing the same procedure until all of the demands have been packed. Therefore the height of each slot is at most $A+\frac{A_{max}}{S_0}$. So in this case we have:

$$P^{\pi}_{\max}\le A+\frac{A_{max}}{S_0}\le\overline{A}+\frac{A_{max}}{\ell} \tag{13}$$

C. Non-Ideal Cases

In this case $t=1$ is NOT achievable, so the constraints on ℓ and r are as follows:

$$0<\ell<r<1 \qquad\text{and}\qquad \left\lceil\frac{1}{r}\right\rceil>\frac{1}{\ell}$$

These conditions imply that $\lfloor\frac{1}{r}\rfloor=\lfloor\frac{1}{\ell}\rfloor$. Using Lemma 2, we have: $Z^*=r\cdot\lfloor\frac{1}{r}\rfloor<1$.

In this case the chosen policy π is as follows: We simply divide the time interval $[0,Z^*]$ into $K_0=\lfloor\frac{1}{r}\rfloor$ non-overlapping time slots with equal length r and leave the remaining part of the time interval, i.e. $[Z^*,1]$ *unscheduled*. Now we can use the same approach used in Subsection IV-B, while here $s_i=S_0=r,\quad i=1,\ldots,n$ and $K_0=\lfloor\frac{1}{r}\rfloor$ and also the threshold value is $\frac{A}{Z^*}$. Then in this case we have:

$$P^{\pi}_{\max}\le\frac{A}{Z^*}+\frac{A_{max}}{r}\le\overline{A}+\frac{A_{max}}{\ell} \tag{14}$$

V. Proof of Theorem 1

Proof: Lemma 3 showed that $\overline{A}\le P_{opt}$, which proves the lower bound part of Theorem 1. In addition in Section IV, for each set of demands $\mathcal{N}$ with pair (ℓ,r), resulting in ideal, near-ideal, non-ideal cases, we introduced a policy π with performance $P^{\pi}_{\max}\le\overline{A}+\dfrac{A_{max}}{\ell}$ which results in:

$$P_{opt}\le\overline{A}+\frac{A_{max}}{\ell} \tag{15}$$

which proves the upper bound part of Theorem 1. Therefore combining inequalities (7) and (15) gives us: $P_{opt}\in\left[\overline{A},\ \overline{A}+\frac{A_{max}}{\ell}\right]$ which proves Theorem 1. ■

VI. Computational Results

Algorithm 1, which is called "*Power Strip Packing*" algorithm, summarizes all the steps of packing the demands. Clearly the running time of this algorithm is linear in n, the number of demands. As we discussed in Section IV, we showed for this algorithm we have:

$$P^{PSP}_{\max} \le \overline{A} + \frac{A_{\max}}{\ell} \le P_{opt} + \frac{A_{\max}}{\ell} \tag{16}$$

In obtaining the last inequality in (16) we used Inequality (7) in Lemma 3. So as it was mentioned in Subsection I-C, using the language of [11] and [13], it corresponds to performance ratio being exactly 1. Therefore this algorithm is asymptotically optimal for all different cases. Furthermore as we can see in Algorithm 1, this algorithm is linear time.

We can achieve better performance in terms of flatter $P(t)$ for $t \in [0,1]$ and get lower P_{max}, albeit at the expense of increasing the number of operations. For example the greedy algorithm works by ordering the demands by non-increasing areas (A_i) and select $s_i = S_0$ for $i = 1, \cdots, n$ exactly the same as that of in PSP Algorithm 1 (and hence $d_i = \frac{A_i}{S_0}$). Then start with the first (i.e. largest) demand and put it in a slot with minimum total height and continue this for subsequent demands until packing all of the demands. Note that when greedy algorithm place a new demand in a slot, it is not possible that the height of every slot being greater than $\frac{A}{Z^*}$, otherwise the total area $\frac{A}{Z^*} \times Z^*$ exceeds A, the sum of the demands. Therefore the height of each slot is at most $\frac{A}{Z^*} + \frac{A_{max}}{S_0}$. So the inequalities (13) and (14) are still valid and hence it is also asymptotically optimal. However the running time is $O(n^2)$.

Figures 4 and 3 illustrate the outputs of the presented algorithms (Algorithm 1 and the greedy algorithm) for different values of $|\mathcal{N}|$, the number of demands, in near ideal cases and non-ideal cases respectively. In these figures the demands, i.e. A_i's, are independent and identically uniformly distributed in the interval $[0, \ell]$. For each value of $|\mathcal{N}|$, each of the algorithms is performed 30 times and then the mean value is depicted as the corresponding $P_{\max}$ of each algorithm. Furthermore for some values of $|\mathcal{N}|$ the corresponding 0.95% confidence intervals are shown in these figures.

As it can be seen from these figures, in both cases, with or without ordering always we have: $P_{max} \le \frac{A}{Z^*} + \frac{A_{max}}{\ell}$. However as we can see in these figures, ordering can improve the performance of the algorithm albeit at the expense of increasing the number of operations.

Algorithm 1 Power Strip Packing algorithm

```
INPUT $\ell$,$r$, Demands $A_i$ $i = 1, \dots, n$
OUTPUT $P_{max}$,$H_{max}$ and Scheduling policy
$A = \sum_{i=1}^{n} A_i$, $A_{min} = \min_i\{A_i\}$, $A_{max} = \max_i\{A_i\}$
if $(r \ge 1)$ or $(\ell \le \frac{A_{min}}{A} \le \frac{A_{max}}{A} \le r)$ then
    Perform Ideal Scheduling (Subsection IV-A)
    $P_{max} = H_{max} = A$
else
    if $\{\lceil \frac{1}{r} \rceil \le \frac{1}{\ell}\}$ then $K_0 = \lceil \frac{1}{r} \rceil$ and $S_0 = \frac{1}{K_0}$ and $Z^* = 1$
    else $K_0 = \lfloor \frac{1}{r} \rfloor$ and $S_0 = r$ and $Z^* = r \cdot \lfloor \frac{1}{r} \rfloor$
    $Threshold = \frac{A}{Z^*}$
    Fill each slot until its height exceeds the $Threshold$, then continue with next slot
    $P_{max} = H_{max} = \max_j\{$height of $j$-th slot , $j = 1, \dots, K_0\}$
end if
```

VII. Conclusion

In this paper considering a problem of supplying electricity to malleable demands, we introduced off-line Power Strip Packing (PSP) problem and showed that using a linear time algorithm will result in asymptotically optimal performance. One may extend this problem such that different demands can have different scheduling bounds. Another (and may be more important) extension to this problem is On-line Power Strip Packing where the goal is serving the arriving demands with different stochastic characteristics such as energy demands, scheduling bounds (ℓ, r), arriving times and the deadline for scheduling each of them.

Appendix A
Proof of Lemma 1

Proof: Suppose that there is a sequence s_j, $j = 1, ..., q$ satisfying $\ell \le s_j \le r$. Now Consider the following definition: $s' \triangleq \frac{1}{q}\sum_{j=1}^{q} s_j = \sum_{j=1}^{q} \frac{1}{q} \cdot s_j$ (Clearly $\ell \le s' \le r$). Then w is achievable if and only if there exist an integer value q and a real value s' such that $\ell \le s' \le r$ and $w = q.s'$. So we have:

$$s' = \frac{w}{q} \Longleftrightarrow \ell \le \frac{w}{q} \le r$$
$$\Longleftrightarrow \frac{w}{r} \le q \le \frac{w}{\ell} \Longleftrightarrow \left\lceil \frac{w}{r} \right\rceil \le \frac{w}{\ell}$$

Therefore the sequences s_j, $j = 1, ..., q$ satisfying $\ell \le s_j \le r$ exist if and only if $\lceil \frac{w}{r} \rceil \le \frac{w}{\ell}$. ■

Appendix B
Proof of Lemma 2

Proof: Instead of using $\sum_{i=1}^{q} s_i$, we just use its equivalent value $q \cdot s'$, where $s' = \frac{1}{q}\sum_{i=1}^{q} s_i$. Suppose that there exists a value $v = q \cdot s'$ such that $w^* < v < w$ which means $q \cdot s' > h \cdot \lfloor \frac{w}{h} \rfloor$(Note that $q \cdot s' \ne w$ because w is not achievable). We know that: $s' \le h$. Combining these two inequality results in: $q > \lfloor \frac{w}{h} \rfloor$, which implies that:

$$\left\lceil \frac{w}{h} \right\rceil \le q \tag{17}$$

On the other hand $\ell \le s'$, so we have $q \cdot \ell \le q \cdot s' < w$ which results in:

$$q < \frac{w}{\ell} \tag{18}$$

Combing the inequalities (17) and (18) results in $\lceil \frac{w}{h} \rceil < \frac{w}{\ell}$, which contradicts the fact that w is not achievable (Lemma 1). Therefore the closest point to w which is also smaller than w is $w^* = h \cdot \lfloor \frac{w}{h} \rfloor$. ■

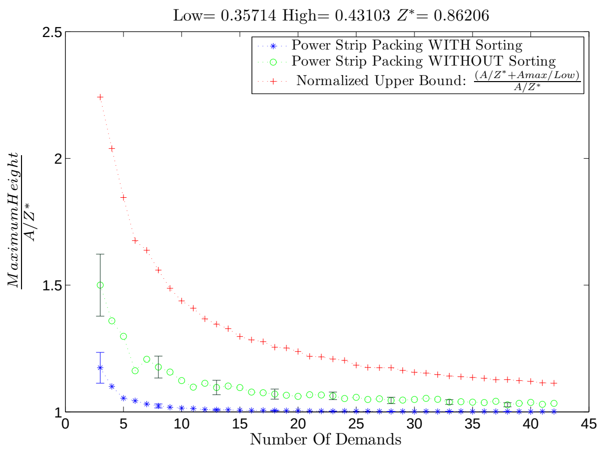

Fig. 3: Comparing performances of the algorithms for Non-Ideal cases with $\frac{A}{Z^*} + \frac{A_{max}}{\ell}$ for different number of demands, where $\ell = 0.3571$ and $r = 0.43103$ and A_i's are independent and identically uniformly distributed in the interval $[0, \ell]$.

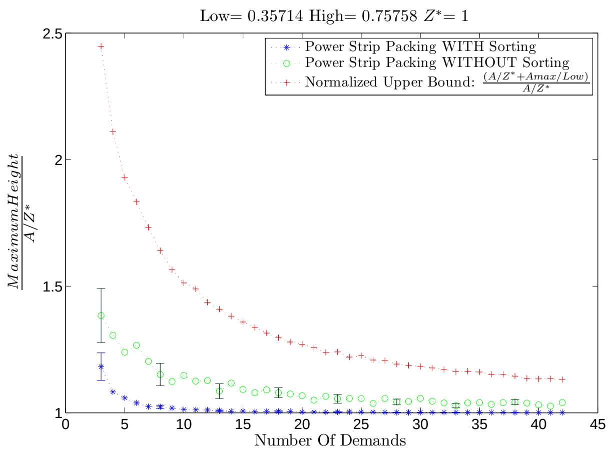

Fig. 4: Comparing performances of the algorithms for Non-Ideal cases with $\frac{A}{Z^*} + \frac{A_{max}}{\ell}$ for different number of demands, where $\ell = 0.35714$ and $r = 0.75758$ and A_i's are independent and identically uniformly distributed in the interval $[0, \ell]$.

Appendix C
Proof of Lemma 4

Proof: To prove Lemma 4 we will show that for any FSP policy θ with P^{θ}_{max}, there exists a filling θ^* with $P^{\theta^*}_{max}$ such that $P^{\theta^*}_{max} \leq P^{\theta}_{max}$. In θ, first consider an interval $\mathcal{T} = [t_1, t_2]$ such that $P(t) = P^{\theta}_{max} \quad \forall t \in \mathcal{T}$. Now suppose there is a gap in $\mathcal{T}$ such that the length of this gap is more than the length of a narrow rectangle completely placed in $\mathcal{T}$. Then this rectangle can be put in that gap while keeping the $P(t) = P^{\theta}_{max}$ in $\mathcal{T}$. Therefore we assume that $\mathcal{T}$ doesn't contain this kind of gaps. So in θ in every interval $\mathcal{T}$ such that $P(t) = P^{\theta}_{max} \quad \forall t \in \mathcal{T}$ any rearrangement of the narrow rectangles does not increase P^{θ}_{max}. Now we try to pick a rectangle scheduled in the interval $\mathcal{T}$ (partly or completely) and put it in an *unfilled* row. This results in a new scheduling θ' with $P^{\theta'}_{max} \leq P^{\theta}_{max}$. Now we repeat this procedure for θ' and continue doing it until we get a scheduling θ^* in which every row becomes filled (maybe except for the last row), so θ^* is a filling with $P^{\theta^*}_{max} \leq P^{\theta}_{max}$. ■

Appendix D
Proof of Lemma 5

Proof: With respect to the conditions $0 < \ell \leq r < 1$ and $\lceil \frac{1}{r} \rceil > \frac{1}{\ell}$, we conclude that $t = 1$ is not achievable. Now define $K_0 = \lfloor \frac{1}{r} \rfloor = \lfloor \frac{1}{\ell} \rfloor$. So we have: $(K_0 + 1) = \lceil \frac{1}{\ell} \rceil > \frac{1}{\ell}$, which means: $(K_0 + 1) \cdot \ell > 1$ and hence: $1 - K_0 \cdot \ell < \ell$. The last inequality shows that there is no space to add even the smallest narrow rectangle (i.e. $s_i = \ell$) to the point $K_0 \cdot \ell \leq 1$ and since $\ell \leq s_i$, K_0 is the largest number of rectangles which can fill a row and don't exceed $t = 1$. On the other hand from Lemma 2, $Z^* = K_0 \cdot r$ is the largest achievable value in $[0, 1]$, i.e. $K_0 r \leq 1$ and $1 - K_0 r \leq \ell$. since $s_i \leq r$, K_0 is the smallest number of rectangles which can fill a row such that there is no space to add a new rectangle to this row. Therefore K_0 is the exact number of rectangles in each filled row. ■

References

[1] L. D. Kannberg, M. C. Kintner-Meyer, D. P. Chassin, R. G. Pratt, L. A. DeSteese, J. G.and Schienbein, S. G. Hauser, and W. M. Warwick, *GridWise: The Benefits of a Transformed Energy System.* Pacific Northwest National Laboratory under contract with the United States Department of Energy. http://arxiv.org/pdf/nlin/0409035, 2003.

[2] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power and Energy Magazine*, vol. 7 , Issue 2, pp. 52–62, 2009.

[3] I. Bayram, G. Michailidis, M. Devetsikiotis, S.Bhattacharya, A. Chakrabortty, and F. Granelli, "Local energy storage sizing in plug-in hybrid electric vehicle charging stations under blocking probability constraints," in *IEEE SmartGridComm 2011 Track architectures and models*, 2011, pp. 78 –83.

[4] N. Ruiz, I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," *IEEE TRANSACTIONS ON POWER SYSTEMS*, vol. 24, No. 2, pp. 959–966, May 2009.

[5] C. Triki and A. Violi, "Dynamic pricing of electricity in retail markets," *A QUARTERLY JOURNAL OF OPERATIONS RESEARCH*, vol. 7, No. 1, pp. 21–36, March 2009.

[6] A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE TRANSACTIONS ON SMART GRID*, vol. 1 , Issue 3, no. 2, pp. 320–331, 2010.

[7] S. Caron and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the smart grid," in *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010.

[8] I. Koutsopoulos and L. Tassiulas, "Control and optimization meet the smart power grid: Scheduling of power demands for optimal energy management," in *Arxiv preprint: http://arxiv.org/abs/1008.3614v1*, 2010. [Online]. Available: http://arxiv.org/abs/1008.3614v1

[9] J. Le Boudec and D. Tomozei, "Satisfiability of elastic demand in the smart grid," in *Arxiv preprint: http://arxiv.org/abs/1011.5606v2*, 2011. [Online]. Available: http://arxiv.org/abs/1011.5606v2

[10] E. Coffman, M. Garey, D. Johnson, and R. Tarjan, "Performance bounds for level-oriented two-dimensional packing algorithms," *SIAM Journal on Computing*, vol. 9, Issue 4, pp. 808–826, 1980.

[11] C. Kenyon and E. Rémila, "A near-optimal solution to a two-dimensional cutting stock problem," *Mathematics of Operations Research*, vol. 25, No. 4, pp. 645–656, 2000.

[12] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, Issue 2, no. 2, pp. 241–252, 2002.

[13] K. Jansen, "Scheduling malleable parallel tasks: an asymptotic fully polynomial time approximation scheme," *Algorithmica*, vol. 39, Number 1, pp. 59 – 81, 2004.

[14] N. Karmarkar and R. Karp, "An efficient approximation scheme for the one-dimensional bin-packing problem," in *23rd Annual Symposium on Foundations of Computer Science (SFCS '08)*, 1982.